

CLAIMS

1. (Previously Presented) A method for representing field structures in a markup language document, comprising:
 - inputting an application document that has been generated by a word-processing application that uses a file format that is specific to the application, wherein the file format is in a non-markup language format that is native to the application and the file format comprises unique properties for describing fields within the document, wherein the unique properties are defined by the application;
 - determining one or more unique properties corresponding to a field that relates to at least one section of the application document;
 - determining whether the field is a complex field or a simple field;
 - mapping the determined properties of the field into at least one of a markup language element, an attribute, and/or a value, wherein the field is designated with a simple field markup language element when the field is determined to be a simple field; and
 - storing the mapped properties of the field in the markup language document whereby applications different from the application can understand the mapped field properties stored in the markup language document.
2. (Canceled).
3. (currently amended) The method of Claim 1 [2], wherein an instruction portion of the field comprises at least one of richly formatted content and embedded additional fields when the field is a complex field.
4. (currently amended) The method of Claim 1 [2], wherein an instruction portion of the field excludes richly formatted content and embedded additional fields when the field is a simple field.

5. (Previously Presented) The method of Claim 1, wherein the simple field markup language element used to designate the field includes a fldSimple element when the field is determined to be a simple field.

6. (Previously Presented) The method of Claim 1, wherein the application itself includes file information among the contents of the document without using text supplied by a user.

7. (Previously Presented) The method of Claim 1, further comprising:
determining properties corresponding to an additional field that relates to at least one section of the application document;
mapping the determined properties of the additional field into at least one of a markup language element, an attribute, and/or a value; and
storing the mapped properties of the additional field in the markup language document.

8. (Original) The method of Claim 1, further comprising:
determining whether properties associated with all fields of the application document have been stored in the markup language document; and
processing further fields when the properties associated with all fields have not been stored in the markup language document.

9. (Original) The method of Claim 1, wherein the properties of the fields stored in the markup language document are understood by an application that understands the markup language when the field is not native to the application.

10. (Original) The method of Claim 1, wherein the markup language document is manipulated on a server to substantially reproduce the field of the application document notwithstanding the presence of an application that generated the markup language document.

11. (currently amended) A computer-readable storage medium for representing fields in a markup language document, comprising:

inputting an application document that has been generated by a word-processing application that uses a non-markup language file format that is specific to the application;

determining properties relating to one or more fields used within the application document, wherein the field comprises unique properties are defined by the application;

determining whether the field is one of a complex field and a simple field;

writing the properties into at least one of a markup language element, an attribute, and a value, wherein the field is designated with a simple field markup language element when the field is determined to be a simple field; and

storing the properties in the markup language document such that the fields of the application document are substantially maintained when the markup language document is parsed by an application that is different from the application used to generate the application document.

12. (currently amended) The computer-readable storage medium of Claim 11, wherein the properties of the fields stored in the markup language document are understood by an application that understands the markup language when the field is not native to the application.

13. (currently amended) The computer-readable storage medium of Claim 11, wherein the markup language document is manipulated on a server to substantially reproduce the field of the application document notwithstanding the presence of the word- processing application that generated the markup language document.

14. (currently amended) The computer-readable storage medium of Claim 11, wherein an instruction portion of the field comprises at least one of richly formatted content and embedded additional fields when the field is a complex field.

15. (currently amended) The computer-readable storage medium of Claim 11, wherein an instruction portion of the field excludes richly formatted content and embedded additional fields when the field is a simple field.

16. (currently amended) The computer-readable storage medium of Claim 11, wherein the simple field markup language element used to designate the field includes a fldSimple element when the field is determined to be a simple field.

17. (currently amended) The computer-readable storage medium of Claim 11, further comprising representing the field with at least one of a fldChar element and instrText element when the field is determined to be a complex field.

18. (currently amended) The computer-readable storage medium of Claim 11, further comprising:
determining properties corresponding to an additional field that relates to at least one section of the application document;
determining whether the additional field is one of a complex field and a simple field;
mapping the properties of the additional field into at least one of a markup language element, an attribute, and a value; and
storing the properties of the additional field in the markup language document.

19. (Previously Presented) A system for representing fields in a markup language document, comprising:
an application that is configured to:
input an application document that has been generated by a word-processing application that uses a non-markup language file format that is specific to the application;
determine properties relating to a field included in at least one section of the application document, wherein the field comprises unique properties are defined by the application;

determine whether the field is one of a complex field and a simple field;
map the properties into at least one of a markup language element, an attribute,
and a value, wherein the field is designated with a simple field markup language element
when the field is determined to be a simple field; ~~and~~
store the properties in the markup language document; and
a validation engine configured to validate the markup language document.

20. (Original) The system of Claim 19, wherein the properties of the fields stored
in the markup language document are understood by an additional application that understands
the markup language when the field is not native to the additional application.

21. (Previously Presented) The system of Claim 19, wherein an instruction
portion of the field comprises at least one of richly formatted content and embedded additional
fields when the field is a complex field.

22. (Previously Presented) The system of Claim 19, wherein an instruction
portion of the field excludes richly formatted content and embedded additional fields when the
field is a simple field.

23. (Original) The system of Claim 19, wherein the markup language document
is manipulated on a server to generate the field of the application document notwithstanding the
presence of the application that generated the markup language document.